Simulation validation: exploring the suitability of a simulation of cell division and differentiation in the prostate

Fiona A. C. Polack, Alastair Droop, Philip Garnett, Teodor Ghetiu, and Susan Stepney

YCCSA, University of York, UK, YO10 5DD
Fiona.Polack@cs.york.ac.uk

Abstract. Individual or agent-based simulation is a potentially important tool for research involving understanding of complex systems. For a research tool to be useful, its use must be understood, and it must be possible to interpret the results of using the tool in the context of the research. This paper discusses issues in the validation of complex systems simulations used as scientific research tools. Specifically, it presents parts of a validation argument developed during construction of a simulation of cells in the prostate – a companion paper describes the models and implementation of the simulation.
Keywords: Complex systems, agent based simulation, validation, argumentation, prostate

1 Introduction

The use of individual-based or agent-based simulation as a scientific research tool requires both good software engineering and robust modelling. For a research tool to be useful, its use must be understood: it must be possible to interpret results from the tool in the context of the research (see, for example, [9, 15, 19]).

This paper is an illustrated discussion of validation – the assurance-related work that parallels development of a simulation. The paper complements Droop et al's description of the modelling and implementation of a prostate cancer cell simulation [4]. This section briefly summarises the simulation development, and existing work on validation of simulations. Subsequent sections present and discuss parts of an argument that the simulation of prostate cells is sufficient for the intended research (the development of prostate cancer). In the discussion that follows, we consider issues identified in the review of the validation argument.

1.1 Modelling prostate cancer

In [4], we note that cancer is known to arise from aberrant interactions among cells. Mutations in a wide range of genes (oncogenes) are known to increase the risk of cancer, but the timing and genetic basis for cancer development is highly variable. This makes the laboratory study of cancers at the expression

level difficult. Modelling collections of cells in a way that allows individual cells to be varied participants, rather than creating models based on population rates and proportions, is an attractive prospect for the study of cancer neogenesis.

In the prostate cancer modelling, our domain expert is a group of researchers from Maitland's Yorkshire Cancer Research lab at the University of York[1]. One of the development team is also a member of this lab: his roles are: to identify biological issues as they arise; to provide background and interpretation of the biology for the developers; and to set up review meetings at which both developers and lab members are represented. In producing the first prototype simulator, there were four major review meetings, at which diagrammatic models were discussed in detail, and changed to better represent the biological understanding of the laboratory researchers. The developer team comprised an implementation expert, two modelling experts and the link-biologist, who has experience of modelling and simulating biological systems. Many of the advances in modelling the prostate cell behaviours occurred in regular discussions among the team.

The prostate cell model simulator is developed in line with the CoSMoS process, a principled approach to modelling and simulation [2, 13]. In CoSMoS, a simulator is a platform on which simulations are run: the simulator is built to support a specific area of research, or *purpose*. The CoSMoS process starts by identification of a domain of interest and of the domain expert(s) who are the primary source of domain information and understanding. A domain model is produced, in close collaboration between developers and domain experts. Early and continuous involvement of domain experts helps to define the purpose, scope and scale of the simulation exercise, the *research context*. From the domain model, developers derive a platform model, a conventional software (or hardware) design. The research context is elaborated with modelling and design decisions, simplifications and assumptions. A simulator is built from the platform model, and is subject to both testing (to establish the quality of the implementation) and calibration (to establish the accuracy of parameters, behaviours etc. using simulation runs initialised to known biological parameters). Throughout the development of the simulator, the research context can be supplemented with a record of sources, assumptions, design decisions, interpretations, etc [2, 13].

In [4], we present the domain and platform models used in developing a prototype implementation. The domain model comprises two levels. The high-level model of cell division and differentiation uses a Petri net (with novel firing semantics). We model the cell-level behaviours with (a) a state diagram for the behaviours of cells in each place and transition in the Petri net; (b) a class diagram (with agent, rather than object semantics) for structure; (c) sequence charts to show how cells are consumed and produced by transitions. These domain models were developed iteratively with expert input, and all were reviewed by the domain experts. From the domain model, a platform model was developed: the structure and design decisions are summarised in [4], which also describes the first prototype of the agent-based simulator implementation in JCSP. There is a clean mapping from domain models to platform models to implementation.

[1] www.york.ac.uk/biology/units/cru/

The CoSMoS process does not end with implementation. Simulations are run on the simulator, to test or develop hypotheses of relevance to the domain of research. The results of a simulation must be interpreted to the domain of research: a simulator is not an exact replica of reality, and data collected from a simulation is about the agents in the simulation context, not about concepts in reality. To support interpretation, CoSMoS develops a results model. The results model draws on the research context, and plays the same role in relation to the simulation as the domain model plays in relation to the domain [2, 13].

1.2 Validation and simulation

The CoSMoS process is a principled approach to modelling and simulation. As part of CoSMoS, we have been investigating validation of simulations – see [6, 7, 12, 13]. Building on traditional simulation development (e.g. [17]) and work in critical systems engineering (e.g. [1, 10]), we present a case for the validity, or suitability, of a simulation as a structured argument over evidence.

The purpose of the validation argument is to present a case, and to expose the case to scrutiny. Validation (of a complex system simulation) can never be absolute; the best we can do is to express the basis on which we believe that the simulation is suitable for its purpose. Polack [12] presents an analysis of the problem of validation and discusses ways in which an argument can be constructed and presented. A key observation is that the validity argument may be incomplete without losing its value: if a simulation is used as primary evidence in research and research publications, there needs to be a thorough and properly documented validation that is exposed to public scrutiny, but if the simulation is used as a subsidiary tool, to help generate hypotheses that can be checked experimentally by the domain experts, then the validation can be partial.

Polack [12] notes the importance of the validation exercise itself, and the mindset that goes with the focus on capturing validation arguments and evidence, in generating a strong collaboration and in raising the profile of simulation as a tool in scientific research.

1.3 Summarising Arguments in GSN

When constructing and presenting an argument, it is useful to provide a diagrammatic summary. This exposes the structure of the argument so that it can be understood and reviewed. We use the Goal Structuring Notation (GSN) [10, 20]. A GSN diagram shows a hierarchy from the top-level claim, through subclaims that support that claim, and eventually to the evidence supporting the claims. The core notation is summarised in Figure 1.

The GSN notation derives from the presentation of safety cases. It is important to understand that the GSN diagram is only a summary of an argument; it is intended to provide an overview and an entry point in to often-complex supporting material. Further, in safety case argumentation, it is the argumentation culture and the safety-case literature that make safety case argumentation

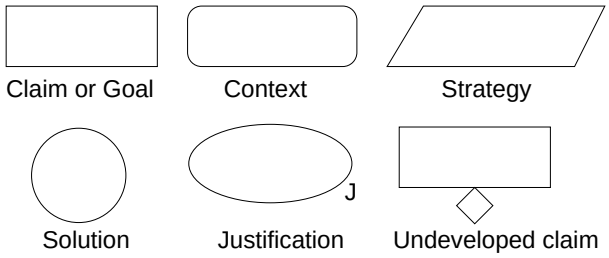

Fig. 1. The basic GSN notations [20, 10]. Note that undeveloped claims in safety case arguments are always subsequently expanded. In our work, an undeveloped claim may be subsequently developed: we indicate claims for which diagrammatic development is included in this paper as filled diamonds.

a powerful tool in the safety field. In using GSN and argumentation in the validation of simulations for research purposes, we similarly aim to influence the culture of development and research. However, there are some significant differences between safety case argumentation and our use of argumentation.

In safety critical systems engineering, a safety case is created and rehearsed by the developers before the argument is constructed and represented in GSN [10]. The safety case must present a complete argument, in which evidence is provided in support of all the claims. Recent work has focused on completeness and clarity of safety arguments, and the need for side-arguments to cover the motivation and justification of the safety case (e.g. [8]). Safety case arguments are constructed to be reviewed by independent authorities, which determine what evidence is and is not acceptable. The argumentation of simulation validity is somewhat different [6]. We do not have a regulator dictating what is acceptable evidence; the argument instead captures the mutual understanding of domain experts and developers; typically it must satisfy both parties, but is not automatically exposed to wider review (though we suggest that wider exposure is important if the subject of validation is high-impact or critical research simulation). As noted above, we do not commit to producing complete arguments, though techniques such as the systematic challenging of models can be useful in exposing assumptions implicit in the development.

Notations such as GSN allow us to capture the structure of an argument; they also support expression of generic arguments and argument patterns – see [6, 18]. From our work on validation arguments [6, 7, 12], we have identified several candidate patterns for simulation validation; in the following, we use several existing patterns as the basis of the prostate cell model validation, and propose the need for several new patterns. As in Weaver's safety case argument patterns [18], however, we find that the value of argumentation to the engineer is in its construction. We only use patterns for (a) high-level structuring and (b) relatively uncontentious and systematic areas such as the statistical analysis of

results. For other parts of the validation, we construct bespoke arguments that draw on the specific experience of each project.

2 The High-level Argument

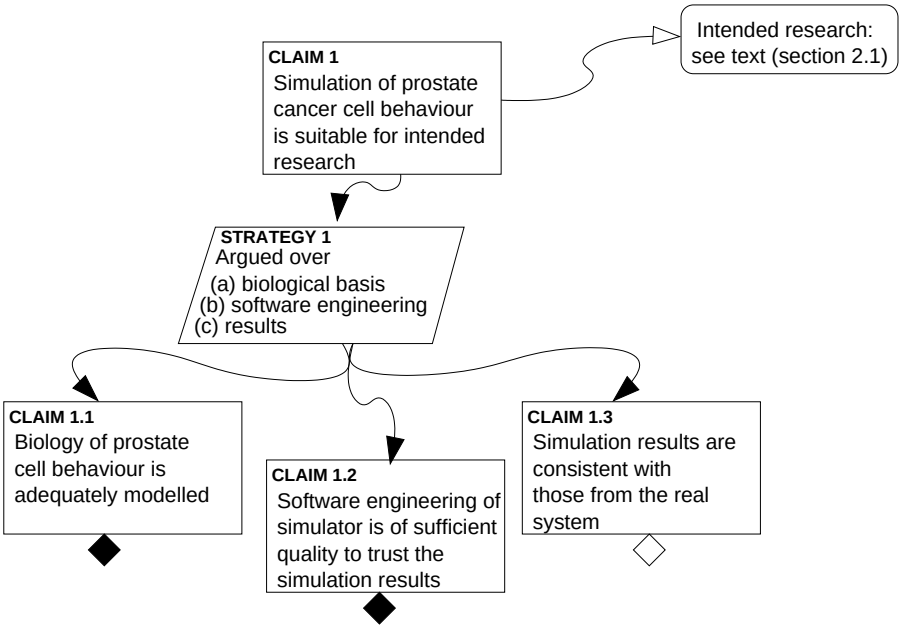

Fig. 2. A top-level argument for the adequacy of the simulation. Undeveloped claims that are expanded diagrammatically in this paper are indicated by filled diamonds.

There are many ways to demonstrate the suitability of a simulation. Here we present an argument that allows us to explore a range of aspects of validation. The high-level argument (Figure 2) uses a pattern identified in [12].

The argument presents a specific claim – that the simulation is suitable for the intended research. This is important: a simulation may not be a good model of its domain, but could be suitable for an intended research goal. More particularly, a simulation that is too faithful to its domain is likely to be almost as complex as that domain, and thus a poor research tool, because it is almost as difficult to understand and manipulate as the natural domain. A connotation of the need to validate against an intended purpose is that if the research goal changes, validation is likely to be invalidated. We cannot use the prostate cell simulator to simulate, say, a breast cell model without identifying new domain experts, revisiting the research context, the domain model, and the validation work.

Here, the claim that the simulation is suitable for the intended research is addressed using a multi-part strategy: we consider the biological basis of the

simulation, the software engineering quality, and the consistency of the results. There is an implicit obligation to ensure that the arguments under each of these sub-claims do not conflict: for instance, in arguing the adequacy of the biological basis, we do not undermine the case for the quality of software engineering, and *vice versa*. Furthermore, there is an obligation to ensure that issues that relate to more than one claim are adequately addressed: that is, the interaction of claims needs to be considered. Here, it is not sufficient to independently address the claim of consistency of the results: it is easy to create a simulation that can mimic some real data distribution, without the concepts of the simulation bearing any resemblance to the structure and behaviour of reality. We must show that the biology is adequately modelled, that the software has appropriate quality, *and* that the results are consistent with observation of the real system.

2.1 Context: Intended Research

The context of a claim in an argument is used to expand terms and definitions in the claim. In Claim 1 (Figure 2), the intended research needs to be defined. The research concerns cell behaviour in the prostate.

The domain for the research is prostate cancer. There are three specific goals for the prostate cell research of which the simulation is part. This paper relates only to the simulation for the first of the following goals; the later goals motivate the purpose of the first simulation.

1. Develop a model of cell differentiation and division, based on prostate cell populations from laboratory research. The purpose of the model is to replicate observed cell population dynamics, represented as changing proportions of cells in a "normal" prostate.
2. Building on the model of the "normal" prostate, develop simulations that capture known environmental variation and mutation. The purpose of the model is to explore the emergence of cell proportions indicative of cancer (or other prostate conditions).
3. Using these models of normal and cancerous prostate cell behaviours, develop simulation experiments that can be used to test biological hypotheses of cancer development and control.

Understanding the context – and the purpose – of the simulation and the claims to be made about it, focuses the scope, scale and level of the simulation. To develop the model of cell differentiation and division, we need to understand the cell biology at an appropriate level. We need to be able to engineer environmental interaction and we need to be able to monitor the proportions of different cells in the simulated prostate.

2.2 The Top-level Strategy

The top-level strategy in Figure 2 divides the validation process into three elements, each of which can be addressed by a separate argument – although, again, care must be taken to ensure that the separate arguments do not conflict.

The argument that is being made can be summarised as stating that the simulation will be considered suitable for the prostate cancer cell behaviour research if the the biological model is adequate, the software engineering has sufficient quality and the simulation results are consistent with those from the laboratory research. You may disagree with this position; the point is that the researchers (prostate cancer scientists and simulator developers) have made their position explicit.

The three sub-claims arising from the three elements of the top-level strategy are considered here at different levels of detail. Most space is devoted to Claim 1.1, concerning the adequacy of the modelling of the prostate cell behaviour. Claim 1.2, concerning software engineering quality, is discussed briefly. For an expansion of Claim 1.3, the reader is referred to [6], where a similar argument over results is presented, and [12], where Polack presents a generic argument over the results of a simulation.

3 Claim 1.1: the biology of the prostate

No research simulation can be used without an understanding of how it models its intended subject. Ideally, a validation would show an explicit mapping from each concept (structures, objects, behaviours) in the subject to a component or subsystem of the simulation. However, there are always parts of the domain that are not understood well enough to provide such mappings. Further, because the intended research focuses on particular aspects and levels, the domain model simplifies and abstracts from the domain. Also, the demands of computation mean that the platform models and the simulator necessarily include non-domain concepts that have no obvious dual in the domain. The prostate cell model is a complex simulation that abstracts away from low-level detail (biochemistry, physics etc), ignores some concepts (anything other than the identified cells and cell behaviours), and simplifies other concepts (cells are represented as a state-specific genome – a pre-defined set of data). The simulator is a digital representation of a continuous model, with imposed synchronisation using the programming idiom of barriers over channels. In practice, it is impractical to try to enumerate all the concessions made in modelling and implementation, let alone understand their connotations. However, to be able to interpret simulation results, and to be able to relate the simulation to the domain, it is important to try to capture assumptions, omissions, abstractions and simplifications, as well as the sources used in developing the simulation.

A problem faced by many simulation developers is that there is disagreement among experts as to the behaviours, or even structures, of a particular subject; working in isolation, a developer has to try to extract a coherent view of the domain. A fundamental aspect of the CoSMoS process is that we rely on the biological input of a specific group of domain experts. The simulator is designed to express one interpretation of a system, and for us, the system interpretation is the view of the designated domain experts.

To address the claim that the biology of prostate cell behaviour is adequately modelled (Claim 1.1), two strategies are proposed. Note that in safety argumentation, such multiple strategies may also be used. The strategies present complementary approaches (such as use of various recognised software engineering techniques with different strengths). The strategies are pursued through sub-claims to evidence, and the safety case depends on the cumulative evidence from all the strategies: the safety case is a balance of evidence across the complementary areas. Here, however, the two strategies are both needed to address the claim; they represent different parts of the argument rather than pointers to complementary evidence.

Strategy 1.1.1 addresses the biological concepts, and argues that the identified cell types and transitions are modelled adequately. Strategy 1.1.2 argues that the scope and scale of the simulation are adequate. The claims that follow from these strategies are now briefly considered. We then address some of the issues that arise across the claims, identifying some of the common problems of simulating complex biological domains and of validating such simulations.

3.1 Claims relating to biological concepts

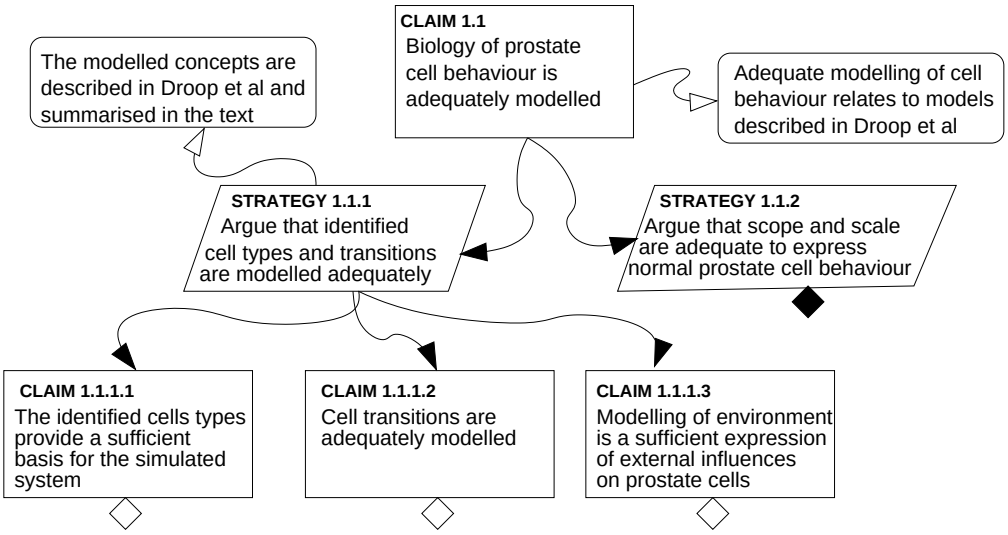

Fig. 3. Claiming adequate modelling of prostate cell behaviours (as described in [4]). Undeveloped claims that are expanded diagrammatically in this paper are indicated by filled diamonds.

The development of Claim 1.1 under Strategy 1.1.1 is shown in Figure 3. The context of the strategy (and the subsequent sub-claims) refers to Droop et al [4], which presents the domain and models of cell structures, behaviours

and transitions. In relation to the argument of suitability of the simulation, the important issue is that all the details of these models were agreed between the developers and the domain experts in the series of review meetings noted above.

Because there is a record of project meetings, and the sources of information are identifiable, we could expand each sub-claim to explore the modelling adequacy of each type of cell, the cell structure and behaviours, and the overall system (described in [4]). We do not present the arguments here for brevity.

The key modelling decisions in the development of the prostate cell simulation are the identifications of cell types and transitions and the environmental factors. Where cells have distinct identity in the prostate, this is straightforward (though, in this respect, the prostate presents an unusually coherent biological subject): the system modelled in the prototype simulation considers the behaviours of stem cells, transit amplifying cells, and committed basal cells. Each type of cell has different responses to the environment, and different behaviours over time, as the cell matures or undergoes mutation. Furthermore, each transition from one cell type to another is determined by environmental and internal factors that differ from cell to cell. (The identified behaviours are captured diagrammatically in [4].) A crucial modelling decision is to represent cell division in two parts: a cell (stem cell or transit amplifying cell) divides in to daughter cells. Each daughter cell then differentiates. This allows careful modelling of the biological options in division and differentiation: for instance, a transit amplifying cell that divides can give rise to two transit amplifying cells, two committed basal cells, or a transit amplifying cell and a committed basal cell. Each transition to or from the daughter cell has been analysed and designed to take account of internal and external (environmental) factors, as well as available biologically-derived rates and probabilities. In the review meetings, each element of these models of division and differentiation has been checked with domain experts: the record of this validation could easily be added to the argument of simulation suitability if the domain experts wished to present the simulation results without independent confirmation through laboratory experimentation.

3.2 Claims relating to scope and scale

The issues relating to cells that need to be explored under the Strategy 1.1.1 are interrelated with issues of scope and scale addressed under Strategy 1.1.2 – the cross-cutting issues are considered further in Section 3.3, below.

In Figure 4, two sub-claims relating to scope and scale are shown. Claim 1.1.2.1, that the numbers of cells in the simulation are sufficient for biologically realistic behaviour to emerge, is not elaborated: the strategy for this claim would be to conduct comparisons of the numbers and proportions of cells in the model. We would investigate whether we need biological-scale numbers, or whether it is sufficient to simply to initialise a simulation with biologically-realistic proportions of each type of cell. This is an important issue of scale: it is known that emergent behaviours do not arise in systems that have fewer than a critical number of components (cells), but it is not usually possible to determine what the critical numbers are. Additionally, the intended research (section 2.1) will

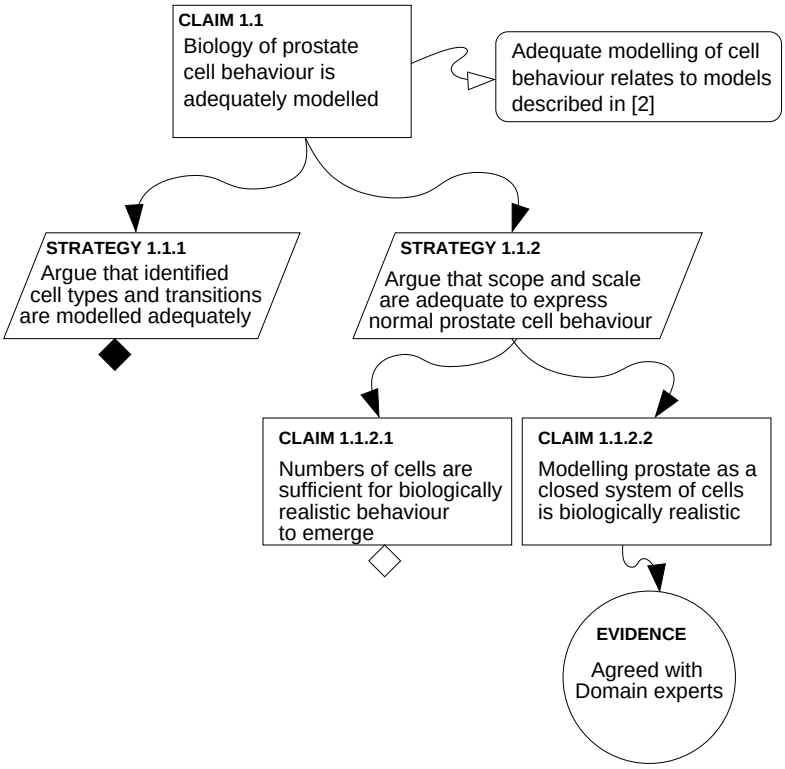

Fig. 4. Claiming adequate scope and scale of prostate cell models. Undeveloped claims that are expanded diagrammatically in this paper are indicated by filled diamonds.

eventually focus on cancer-causing environmental effects and mutations which are low probability events; the agent-based simulation was chosen specifically to allow us to model individual variation, so we need a simulator with sufficient numbers of cells to accommodate low probability events.

In biological systems, it is usually possible to estimate numbers and proportions of cells, and, to some extent, how these values change over time. Biologically-robust data can be used to calibrate a simulator. Simulation experiments can explore the adequacy of scale by initialising the simulator with biologically-derived proportions or numbers of cells: if the simulator is sufficient for the intended research, then running the simulation with either set-up should give biologically realistic later cell counts if the scale and scope of the simulation is appropriate (and if all the other biological elements of the validation are also suitable!). We return briefly to the issues outlined here in Section 5.

Claim 1.1.2.2 demonstrates how a claim can be concluded. Here, the claim, that modelling the prostate as a closed system of cells is biologically realistic, is substantiated by a reference to evidence: that this has been agreed with the domain experts. Clearly, such a statement is open to challenge: what was the

basis of agreement? why is a closed system considered to be appropriate by the domain experts? We return to the discussion of this claim and evidence in section 5. The issue, in itself, is important: many open biological systems are simulated by closed systems, but the implications of closure are not considered. The research context needs at least to identify that this is an issue, so that it is flagged as an "unknown effect" when simulation results are considered.

3.3 Cross-cutting issues of biological simulation and validation

Scope and scale are intrinsically linked with issues of adequate modelling. Cross-cutting issues also relate to the ways in which the simulator can be used. To be amenable to validation, a simulator must be created for a specific purpose, and cannot be used to support simulations with other purposes unless it is subject to re-validation (section 2).

The cell types and transitions identified for the simulation (and subject to claims under Strategy 1.1.1) are based on the domain experts' understanding of the important cell divisions and differentiations that take place over space and time in the prostate. The information that the domain experts give to the developers is based on the understanding of the biology of the prostate in this laboratory, including any theoretical understanding that underpins this laboratory's work. The simulation constructed for the prostate is not general: a laboratory with a different theoretical standpoint, or a different interpretation of the biology, would not be able to test their hypotheses on this simulation. Furthermore, the simulator expresses a scale and scope that matches the scope of the laboratory research of the domain experts. The tie-in of the simulator to the domain expert view means that, if the results of simulation are to be published to the wider prostate cancer community, it particularly important to record the biological and theoretical basis of the simulation.

The specific domain scope of the simulator also defines the levels of abstraction. In the biological domain, the cells (naturally) have complex structure and biochemistry. In identifying the cell concepts to include, the domain experts work with the developers to make appropriate abstractions and simplifications; the scale of abstraction is, again, intrinsically linked with the biological modelling issues. Whilst it would be an interesting challenge to construct a simulator that modelled the biochemistry of signalling as well as the cell division and differentiation, this would take us away from the scientific purpose and motivation for this simulation exercise. For the intended purpose of this simulator, it is not necessary to know how cells emit and receive chemical signals; it is sufficient to know that cells affect the environment, and that the environment affects cells, through particular interactions and behaviours. The model of the environment must include the appropriate parameters to implement this interaction, guided and checked by the domain experts. If the argument that the simulator is sufficient were extended, the validation of each of these areas would be added; we might also add a separate set of claims relating to the cross-validation issues.

Another issue that commonly arises in considering both the details of the biological modelling and the scale and scope of a simulation is the representation

of space. In the prostate, cells take up volume, and it is postulated that some of the effects observed in the biological system are related to crowding in the different spatial zones of the prostate. It is certainly the case that division of cells is related to the availability of space. Design decisions have to be made as to what to do about space in the simulation. For example, we could construct a 3D simulation in which space is explicit – this allows realistic visualisation, as well as a natural interpretation of crowding. In validation terms, there is a direct mapping between biological space and simulator space, and this should make it easy to interpret simulator results on spatial distribution and dynamics. Unfortunately, the realistic 3D space model is not always easy to relate to the biologically-observable data, since there are only very limited ways in which researchers can observe the internal dynamics of a real prostate – this is one of the motivations for turning to agent-based simulation as a research tool.

In the prostate cell simulation development, we are currently using an easier simulation approach, which provides a surrogate for crowding. The domain experts can provide data on the size and cell behaviours of the different spatial areas of the prostate. We can use this data to estimate typical capacities, and derive upper limits to cell division as part of the environment.

Space modelling is one of many issues for which we would like to explore generic argument patterns to guide developers and domain experts in the development of simulations that are amenable to validation. Note that the decisions about how to model space are in the remit of the developers rather than the biological domain experts. The domain experts may need to understand whether space is implemented directly or through the surrogate of division limits, in order to understand features of the results.

As an aside, the consideration of space leads to a nice potential experiment combining simulation and laboratory science: use biological data to develop and drive the surrogate-crowding model; compare the results to reality and tune the simulation to normal prostate behaviour – this relies on the equivalence of the observable data from reality and the simulation. Now, create a 3D simulation with "real" space, and tune this to have the same dynamic characteristics as the surrogates-crowding model; identify hypotheses on the 3D simulation that can be tested in the laboratory, to advance understanding of system dynamics.

4 Claim 1.2: the quality of software engineering

Claim 1.2, in Figures 2 and 5, concerns the quality of the software engineering of the simulation. We cannot easily measure the quality of software, especially where the software implements a complex system; we need alternative ways to develop confidence in the quality of the simulator as an artifact. We could add a justification to the presented argument outlining how the quality of the simulator depends on the quality of the software engineering. This paper is not about the engineering of simulations; however, software engineering quality is important, and there are likely to be some research simulations that are of sufficient importance and criticality that the software engineering validation and its

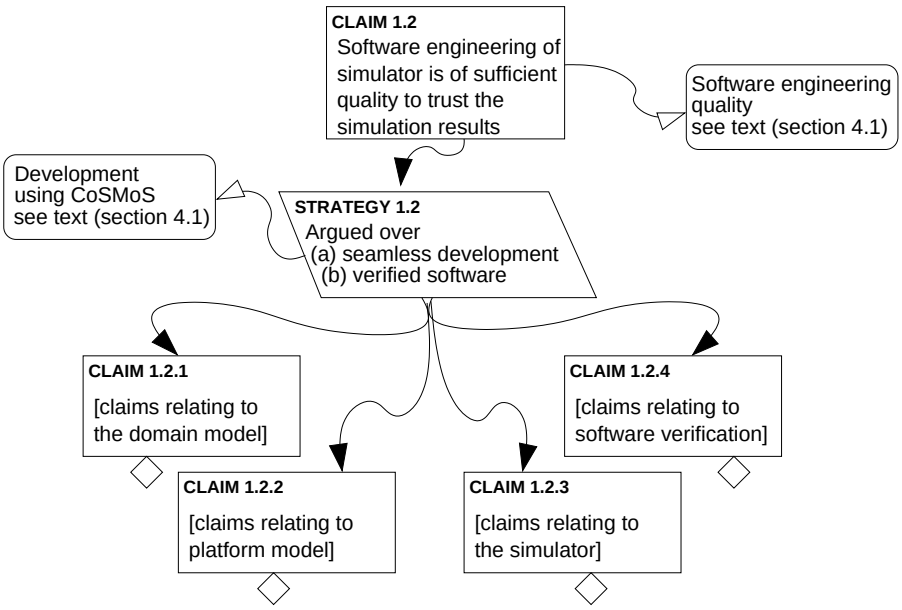

Fig. 5. Claiming Software Quality

presentation would have to be treated with as much rigour as the validation of biological aspects and results: we return to issues of sufficiency in section 5.

Rather than elaborate the software engineering claims here, we discuss the context and strategy in Figure 5 and explore some of the issues that these raise. The sub-claims in Figure 5 are inspired by a validation pattern proposed by Ghetiu et al [7] that focuses on the validation of the development process. This pattern could be developed as a generic pattern for validation of simulations developed in line with the CoSMoS process.

4.1 Issues relating to Software Engineering Quality

Clearly, a claim over the quality of software engineering needs to make some statement about the standpoint of the authors on software engineering and quality. Here, quality is related to techniques and principles used in the development. The strategy for the argument as to the quality of the software engineering (Strategy 1.2 in Figure 5) is to consider seamless development and verification.

Quality of software engineering takes for granted that we have worked out what system we need to construct: that requirements are known and have been appropriately validated with clients. Here, we rely on the CoSMoS process [2, 12] to sort out these early issues: CoSMoS uses the process of domain modelling to cement the relationship between developers and researchers (the clients), to clarify the scope and levels of the simulation, and to determine the purpose of simulation (see section 1.1). CoSMoS also proposes the seamless development

from domain model to platform model and to simulator implementation which gives rise to the proposed Claims 1.2.1 go 1.2.3 in Figure 5. The elaboration of these claims can refer to the principled development guidance of the CoSMoS process: a similar appeal to a particular development method is used when considering the artifact quality in safety case argumentation.

Our potential quality argument thus appeals to the recognised value of seamlessness in software engineering: adhering to the same paradigm and semantics for modelling and the implementation reduces opportunities for misinterpretation. Elsewhere (e.g. [14]), Polack et al. discuss the need for such continuity in simulation development. In [4], we present the multi-model specification for the prostate cell simulator, and describe a reasonably seamless implementation process from the models. Indeed, we amend the semantics of diagrammatic modelling notations to provide a clean mapping to the code design. A complete argument of software engineering quality would systematically present the model mappings and development. The argument would be strengthened if the practical seamlessness of such a development approach were to be supported by techniques model-driven engineering, and specifically by developing and supporting domain specific languages (see [4, 14]).

Any validation of a simulation assumes that the software has been verified: that the system has been tested and shown to work as intended. Testing of complex systems simulations is interesting, since the results are often non-deterministic: multiple runs and statistical analysis of data results are needed to establish whether, with some likelihood, the results of the simulation are in line with those from laboratory experiments. In the validation argument summarised here, there is thus a close link between the software engineering claims and the results claim, Claim 1.3.

For conventional software testing, programming environments (e.g. Eclipse for Java) and built-in debugging and analysis tools are useful, but, as Polack et al note [14], environment support for many paradigms and languages is still primitive. This area is often unsatisfactorily concluded in validation of simulations. Performance and sensitivity analysis can go some way towards identifying problems and mitigating the effects of errors in design and implementation [16]. When simulating a complex system, it is typically unclear whether an observed behaviour of the simulation is genuine or is the result of a bug in the implementation – observed behaviours may even be due to a "mistake" in the design. There can be a fine line between a desired emergent behaviour and the consequences of a bug in a design or a program; the simulation developers as well as the domain experts must be confident that the observed behaviours are "valid", not artifacts of a faulty simulator.

5 Discussion

The argument outlined in this paper was developed after the development of the domain and platform models, but before completion of the simulator, or initialisation and running of any simulation. A validation argument is a living

argument, in two senses. Firstly, as we conduct simulations and analyse results, we use the research context built up in the development and validation process to interpret the results. Secondly, by using the simulator and conducting *in silico* experiments, we learn about the domain, and about the simulator, and can expand the validation appropriately. In this sense, the argument is a snapshot relating to the simulation project on the day that it was produced. If we want to publish the simulation results, and expose the simulator to community evaluation, we need to ensure that the validation argument is up-to-date and matches exactly the published version of the results.

To show how an argument can evolve, we now consider some of the issues that arose when the authors met to review the argument presented in this paper, during the writing of the paper: that is, after the domain experts and developers had agreed on the model and the detail of the model to be implemented.

The most significant issue identified relates to Claim 1.1.2.2 (Figure 4). The claim states that *Modelling prostate as a closed system of cells is biologically realistic*. The first observation is that we have not explained what we mean by a closed system of cells: this can be addressed by adding a context explanation. The context is that that the agreed model comprises a precise set of cell types, their division and differentiation, and a limited (finite) set of environmental influences: the domain experts agreed that we do not need to model blood flow, nutrient supply and other biological fluxes in order to develop a simulation of the prostate cell model that is suitable for the intended research (Claim 1). We can also strengthen the presented argument by elaborating the evidence either: by extending the represented argument, or by providing a justification that explains why this evidence is sufficient.

In reviewing Claim 1.1.2.2, we therefore clarified what had been agreed in relation to system closure. This prompted the biologist in the review to query the claim: it is well known that no biological system is closed. What are the implications of a closed system model? This important question highlights several specific research questions that we cannot address with this simulator, and thus we are able to more precisely determine the "intended research" for which this simulator is suitable. The model does not include an explicit model of blood vessel formation and capillary bed structure. However, it is known that many tumours develop limited blood-supply (vascularisation), and thus have lower oxygen levels (hypoxia) than non-cancerous tissue. It is possible to target areas of hypoxia for delivery of chemotherapeutic drugs[11]; even in tumours with adequate vascularisation, interventions can be targeted using drugs that interrupt blood vessel formation[3]. Since we cannot model hypoxia, when we use simulation to explore possible interventions in phase 3 of the research, we cannot use the simulator to explore interventions that rely on blood flow or hypoxia.

Moving beyond the immediate connotations of scoping out vascularisation and blood flow, we observe that the simplifications made in modelling always imply hidden assumptions in the modelling of the prostate cell system. The environmental parameters that are included in the model implicitly provide surrogates for things that are not in the model. Thus, it is hard to map from a

biologically-motivated environmental change (such as a chemical signal) to a suitable change in an environmental parameter of the simulator. For a critical simulation, we would need to (a) take steps to identify hidden assumptions; (b) analyse the effect of surrogation by model concepts for concepts not modelled; (c) extend the validation argument accordingly. To date, we have not undertaken such a high-impact simulation study, but we have identified a range of critical systems engineering techniques that can help to challenge models and identify assumptions [14]. Read et al [16] provide some initial consideration of the surrogation problem in calibration and sensitivity analysis, as well as results interpretation.

Turning to the undeveloped claims relating to the quality of the software engineering (Figure 5), the review undertaken in writing this paper identified an important omission: the argument has not addressed the calibration of the simulator. The CoSMoS process defines calibration as a tuning activity [2]. Typically, calibration involves trying to replicate in simulation the normal operation of the real system; parameters, behaviours, initialisation and scale of operation may need to be modified. This fine-tuning exposes assumptions, abstractions and simplifications (in relation to the science and the development of the simulation), as well as validating performance and outputs [5]. Calibration is important as it is relates to the quality of the software product (the simulator), but draws on the biological domain. The calibration must identify "normal" biological scenarios, for which the domain experts can provide suitable data both for initialisation of a simulation and for evaluation of the results. The activity of calibration thus draws on the whole development and documentation of the simulator, and fundamentally relies on the close collaboration built up in the development of the simulator. It is worth noting that the calibration of our simulator in itself achieves the purpose of the first phase of the simulation project: to replicate observed proportions of cells in the normal prostate (Section 2.1). Calibration had not been completed when these arguments were constructed, but it should have been flagged in the argument. We intend to construct a calibration argument pattern to guide this vital activity.

In relation to the un-developed claims of the software engineering quality argument, the review took place before completion of the platform model and prototype implementation described in in [4]. This led the reviewers to explore in more detail what distinguishes a domain model and a platform model: the outcomes and their interpretation in this study are as follows.

1. A domain model typically expresses the whole domain of interest, including emergent behaviours. The emergent behaviours are the desired outcome of the simulator, and are not included in the platform model (they are not implementation concepts). The desired emergent behaviour of the simulator (in this first phase of the project) is a specific dynamic distribution of cell types: this is not expressible in the modelling formats used in the domain model, but needs to be captured. We can extend the GSN argument structure to include hyperlinks to biological scenarios and data that will be used as the benchmark for the biological calibration.

2. The platform model typically includes implementation-oriented devices needed to capture the biological realities, and instrumentation related to visualisation and results-expression. The major devices employed here are (a) the representation of cells as agents (with the appropriate simplifications) that change type as they differentiate; and the introduction of explicit daughter cells. Both were agreed by the domain experts to be a plausible if biologically-unrealistic approaches, and thus the models presented to the domain experts included these features. Subsequent work on the platform model and JCSP implementation has identified further design decisions and instrumentation required to implement the simulator; these need recording and adding in to the software engineering validation (see [4, §7].

A more general point that arose in review of the arguments, which is a concern in safety case argumentation as well, is the importance of the gaps in the arguments: in particular, the need to check that the expressed claims can also address combined issues. As already noted, the adequate modelling of the biology of prostate cell behaviour (Claim 1.1) is not independent of the quality of the software engineering (Claim 1.2), and the consistency of the simulation results (Claim 1.3) is tightly coupled to these other claims. We need to explore the capture and expression of cross-cutting claims, and derive patterns to guide argument construction in these areas.

In the course of reviewing an argument, it is inevitable that people query the detail and the extent of an argument. In safe-systems engineering, such queries have to be taken very seriously and addressed before the safety case is accepted. However, in our more informal use of argumentation, it is sufficient that a consensus is reached on the adequacy of the argument – and, as described above, important observations on the adequacy of the argument are pursued and concluded. In work using the CoSMoS principled approach and informal validation of the sort described in this paper, we have not yet failed to reach consensus, but a notable feature is that the domain experts are often *more* trusting of the simulators than are the simulation developers. This observation needs following up in patterns of guidance for the construction and review of validation arguments in collaborative research.

An important exception to informality of argument construction and consensus would arise if the simulation evidence were to be used un-supported in prostate cancer research and interventions: at this point, the simulation constitutes a critical system, and must be thoroughly tested, validated and understood. We are exploring potential links between validation argumentation and standards for presenting scientific models and results in, for example, ecological domains.

6 Summary and Conclusions

We have presented aspects of the validation of the prostate cell model simulation (see companion paper [4]). The meaning of validation in the context of complex system simulation is summarised, along with the argumentation approach used

here. The validation argument uses both pre-existing patterns and bespoke elements that are devised for this specific simulator and purpose. Further patterns and guidance are proposed as a result of the work presented here.

The argument structure is summarised using GSN, with selected explanations in the accompanying text. The argument represents the mutual understanding of the sufficiency of the simulation at a particular point in the development: it will develop as the project progresses. In discussion, we identify some of the immediate benefits of constructing and reviewing this argument, in terms of things that have been overlooked or inadequately thought through in the simulation, the simulation process and the validation.

In future, we will revisit this argument (in the light of recent completion of the implementation and calibration). The argument will also form a starting point for the development of the next two phases of the research (section 2.1), as well as an important input to the results model, for interpreting the results of subsequent *in silico* hypothesis generation and testing.

More generally, we are seeking to develop tool support for argumentation that can cross-link argument structures, sources, and text elaboration. We intend to use the argumentation approach in support of systematic documentation of simulation-based research.

Acknowledgements

This work is supported by a Program Grant (to N. J. Maitland) from Yorkshire Cancer Research, by TRANSIT (EPSRC grant EP/F032749/1) through the York Centre for Complex Systems Analysis, and by CoSMoS (EPSRC grant EP/E053505/1). We would like to thank the members of the Cancer Research Unit in York for their invaluable input of time and expertise.

References

1. R. Alexander. *Using Simulation for Systems of Systems Hazard Analysis*. PhD thesis, Department of Computer Science, University of York, 2007.
2. P. S. Andrews, F. A. C. Polack, A. T. Sampson, S. Stepney, and J. Timmis. The CoSMoS Process, version 0.1. Technical Report YCS-2010-450, Dept of Computer Science, Univ. of York, 2010. www.cs.york.ac.uk/ftpdir/reports/2010/YCS/453/YCS-2010-453.pdf.
3. D. Bishop-Bailey. Tumour vascularisation: a druggable target. *Current Opinion in Pharmacology*, 9:96–101, 2009.
4. A. Droop, P. Garnett, F. A. C. Polack, and S. Stepney. Multiple model simulation: modelling cell division and differentiation in the prostate. submitted to CoSMoS Workshop, 2011.
5. P. Garnett, S. Stepney, F. Day, and O. Leyser. Using the CoSMoS process to enhance an executable model of auxin transport canalisation. In *Workshop on Complex Systems Modelling and Simulation*, pages 9–32. Luniver Press, 2010.
6. T. Ghetiu, R. D. Alexander, P. S. Andrews, F. A. C. Polack, and J. Bown. Equivalence arguments for complex systems simulations - a case-study. In *Workshop on Complex Systems Modelling and Simulation*, pages 101–140. Luniver Press, 2009.

7. T. Ghetiu, F. A.C. Polack, and J. Bown. Argument-driven validation of computer simulations – a necessity rather than an option. In *VALID*, pages 1–4. IEEE, 2010.
8. R. Hawkins, T. Kelly, J. Knight, and P. Graydon. Safety cases – a new approach to creating clear safety arguments. In C. Dale and T. Anderson, editors, *Advances in Systems Safety: SSS'11*, pages 3–23. Springer, 2011.
9. P. Humphreys. *Extending Ourselves: Computational Science, Empiricism, and Scientific Method*. Oxford University Press, New York, 2004.
10. T. P. Kelly. *Arguing safety – a systematic approach to managing safety cases*. PhD thesis, Department of Computer Science, University of York, 1999. YCST 99/05.
11. S. Kizaka-Kondoh, M. Inoue, H. Harada, and M. Hiraoka. Tumor hypoxia: A target for selective cancer therapy. *Cancer Science*, 94(12):1021–1028, 2005.
12. F. A. C. Polack. Arguing validation of simulations in science. In *Workshop on Complex Systems Modelling and Simulation*, pages 51–74. Luniver Press, 2010.
13. F. A. C. Polack, P. S. Andrews, T. Ghetiu, M. Read, S. Stepney, J. Timmis, and A. T. Sampson. Reflections on the simulation of complex systems for science. In *ICECCS*, pages 276–285. IEEE Press, 2010.
14. F. A. C. Polack, P. S. Andrews, and A. T. Sampson. The engineering of concurrent simulations of complex systems. In *CEC*, pages 217–224. IEEE Press, 2009.
15. F. A. C. Polack, T. Hoverd, A. T. Sampson, S. Stepney, and J. Timmis. Complex systems models: Engineering simulations. In *ALife XI*, pages 482–489. MIT press, 2008.
16. M Read, P. S. Andrews, J. Timmis, and V. Kumar. Towards qualifying the implications of epistemic uncertainty on simulation based experimentation through calibration, uncertainty and sensitivity analysis. *Mathematical and Computer Modelling of Dynamical Systems*, 2011. accepted.
17. R. G. Sargent. Verification and validation of simulation models. In *37th Winter Simulation Conference*, pages 130–143. ACM, 2005.
18. R. A. Weaver. *The Safety of Software – Constructing and Assuring Arguments*. PhD thesis, Department of Computer Science, University of York, 2003. YCST-2004-01.
19. M. Wheeler, S. Bullock, E. Di Paolo, J. Noble, M. Bedau, P. Husbands, S. Kirby, and A. Seth. The view from elsewhere: Perspectives on ALife modelling. *Artificial Life*, 8(1):87–100, 2002.
20. S. Wilson, J. McDermid, P. Fenelon, and P. Kirkham. No more spineless safety cases: A structured method and comprehensive tool support for the production of safety cases. In *2nd International Conference on Control and Instrumentation in Nuclear Installations (INEC'95)*, 1995.